```python
import pdb

def accel(N, x, y, z, ax, ay, az, sx, sy, sz):

    for i in range(N):
        ax[i] = 0
        ay[i] = 0
        az[i] = 0

    pot = 0

    for i in range(N-1):
        #print('thisisi',i)
        for j in range((i+1),N):
            #print(j)
            #print(i,j)
            dx = x[i] - x[j]
            dy = y[i] - y[j]
            dz = z[i] - z[j]

            adx = abs(dx)
            ady = abs(dy)
            adz = abs(dz)

            if adx > .5*sx:
                dx = dx-sx * dx/abs(dx)
                #dx = (sx-adx) #
                #dx = -dx if dx == adx else dx
            if ady > .5*sy:
                #dy = (sy-ady)
                #dy = -dy if dy == ady else dy
                dy = dy-sy * dy/abs(dy)
            if adz > .5*sz:
                #dz = (sz-adz)
                #dz = -dz if dz == adz else dz
                dz = dz-sz * dz/abs(dz)

            ri2=1/(dx*dx+dy*dy+dz*dz)   #1/r^2
            ri4=ri2*ri2
            ri6=ri4*ri2
            ri8=ri6*ri2
            pot=pot+(ri6-1)*ri6
            b=24*(2*ri6-1)*ri8
            #print('STUFF',ax[i],b,dx,b*dx,ax[i]+b*dx)
            #test = ax[i]+b*dx
            ax[i] = ax[i]+b*dx
            #print('what the fuck',ax[i])
            ay[i]=ay[i]+b*dy
            az[i]=az[i]+b*dz
            ax[j]=ax[j]-b*dx
            ay[j]=ay[j]-b*dy
            az[j]=az[j]-b*dz
    pot = 4*pot
    return(ax, ay, az, pot)
```

```python
############################################################
counter = 0

def update(N,x,y,z,vx,vy,vz,ax,ay,az,sx,sy,sz,dt,ek,kinetic,potential):
    global counter

    ek = 0
    hdt = .5*dt
    #hdt2 = hdt*dt
    for i in range(N):
        vx[i]=vx[i]+hdt*ax[i]
        vy[i]=vy[i]+hdt*ay[i]
        vz[i]=vz[i]+hdt*az[i]

        x[i]=x[i]+dt*vx[i]
        y[i]=y[i]+dt*vy[i]
        z[i]=z[i]+dt*vz[i]

    ax, ay, az, pot = accel(N, x, y, z, ax, ay, az, sx, sy, sz)
    #print('acc',ax, ay, az)
    #accel(N,x,y,z,ax,ay,az,sx,sy,sz,pot)
#------------------------- enforce pbc-------
    for i in range(N):
        #x[i]=dmod(x[i]+sx,sx) #modulus
        x[i] = (x[i]+sx) % sx
        #y[i]=dmod(y[i]+sy,sy)
        y[i] = (y[i]+sy) % sy
        #z[i]=dmod(z[i]+sz,sz)
        z[i] = (z[i]+sy) % sz

        vx[i]=vx[i]+hdt*ax[i]
        vy[i]=vy[i]+hdt*ay[i]
        vz[i]=vz[i]+hdt*az[i]

        ek=ek+(vx[i]**2+vy[i]**2+vz[i]**2)


    kinetic.append(.5*ek)
    potential.append(pot)


    counter +=1

    #if counter == 6803: 5957
     #   pdb.set_trace()

    #if abs((kinetic[len(kinetic)-1]+potential[len(potential)-1]) - (kinetic[len(kinetic)-2
        #pdb.set_trace()


    #if abs(total(len(total)) - total(len(total-1)) > .5:


    return(x,y,z,vx,vy,vz)
```